

Firmware and Software Status

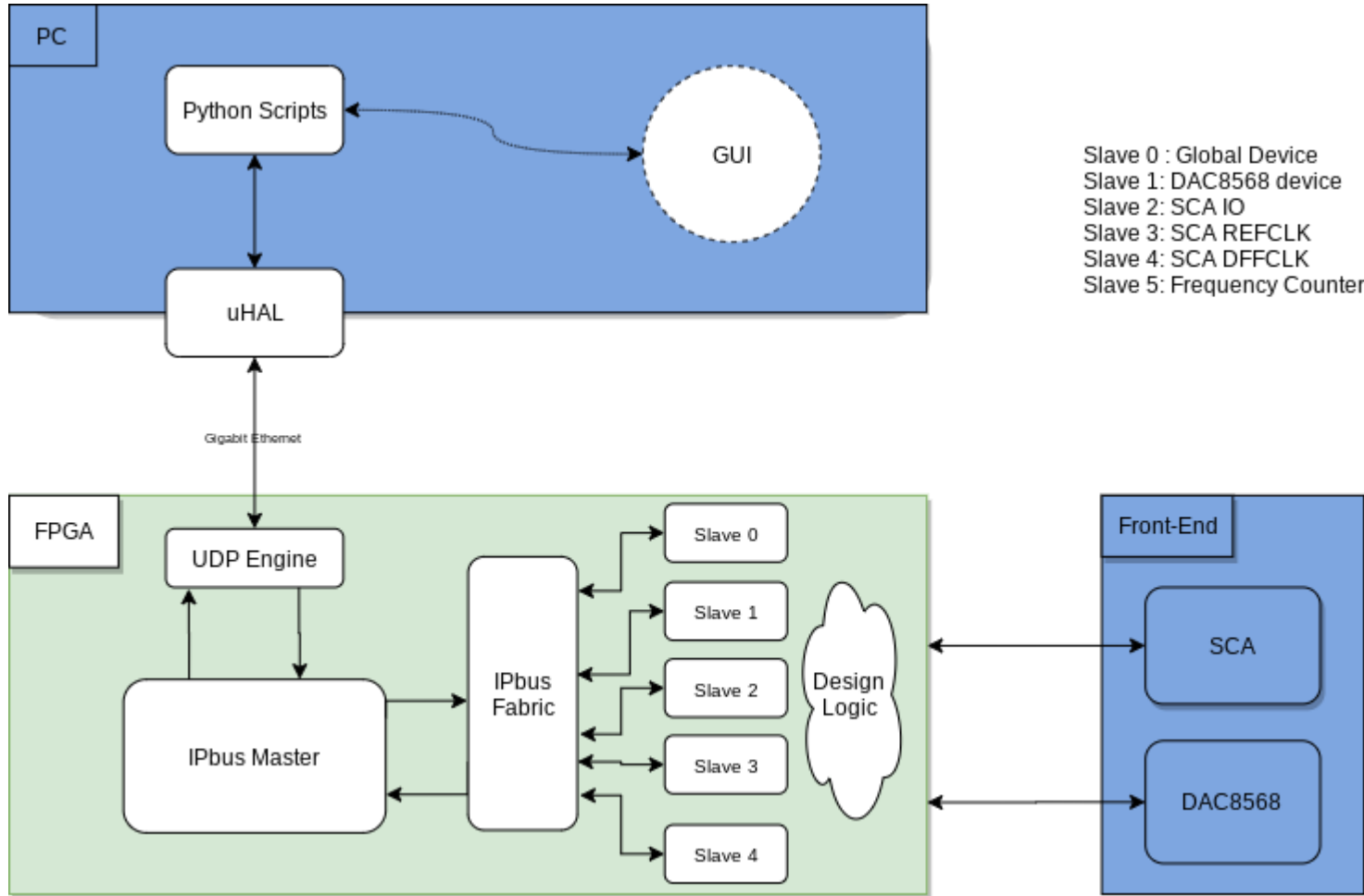
Sheng Dong

s.dong@mails.ccnu.edu.cn

Weekly meeting

Wednesday, July 22, 2020

Test Setup

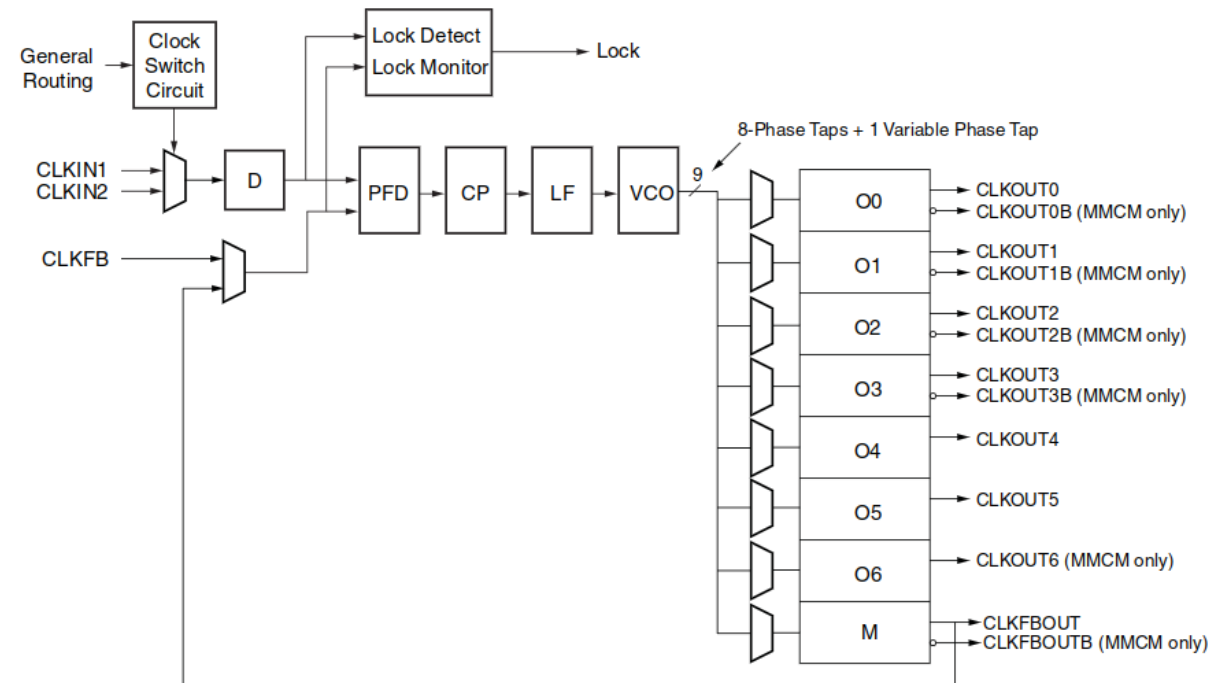


For monitoring and debugging, added a frequency counter slave.



Firmware Design

- IPbus based (<https://ipbus.web.cern.ch/>)
- Fixed IP/Mac address while building firmware, for now they are
 - IP: 192.168.200.16
 - Mac: 020ddba1151
- SCA IO is controlled via IPbus register slaves
- SCA Clocks (REF, DFF) are controlled via IPbus DRP slaves. MMCM and PLL Dynamic Reconfiguration (XAPP888)



x888_01_122211

Figure 1: MMCM and PLL Block Diagram

$$F_{VCO} = F_{CLKIN} \times \frac{\tilde{M}}{D}$$

$$F_{OUT} = F_{CLKIN} \times \frac{M}{D \times O}$$

Software

Python Scripts:

```
|-- lib
|   |-- __init__.py
|   |-- dac8568_device.py
|   |-- freq_ctr_device.py
|   |-- global_device.py
|   |-- sca_device.py
|-- sca_ctrl.py
```

Usage:

Just run **sca_ctrl.py**

By editing the parameters in this script, like IO value, clocks' parameters, we can do various tests.

```
21 if __name__ == '__main__':
22     device_ip = "192.168.200.106"
23     device_uri = "ipbusudp-2.0://" + device_ip + ":50001"
24     # address_table_name = sys.argv[2]
25     address_table_name = "../etc/address.xml"
26     address_table_uri = "file://" + address_table_name
27
28     uhal.setLogLevelTo(uhal.LogLevel.WARNING)
29     hw = uhal.getDevice("HappyDaq.udp.0", device_uri, address_table_uri)
30
31     global_dev = GlobalDevice(hw)
32     sca_dev = ScaDevice(hw)
33     freq_ctr_dev = FreqCtr(hw)
34
35     ## Soft reset
36     global_dev.set_soft_rst()
37
38     ## Set Sca Clocks
39     ## parameters: Do, M, D, clk_in = 125MHz
40     ## Freq: (clk_in * M)/(DO * D)
41     sca_dev.set_clock(1, 1, 1, "clk_ref")
42     sca_dev.set_clock(1, 1, 1, "clk_dff")
43
44     ## Frequency counter
45     freq_ctr_dev.get_chn_freq(0)
46
47     ## Set Sca IO
48     sca_dev.set_bit0(True)
49     sca_dev.set_bit1(True)
50     sca_dev.start(True)
51     sca_dev.dff_enable(True)
52     sca_dev.trigger(True)
53     time.sleep(0.001)
54     sca_dev.trigger(False)
55
```

The test environments

This readout(control?) system is developed on:

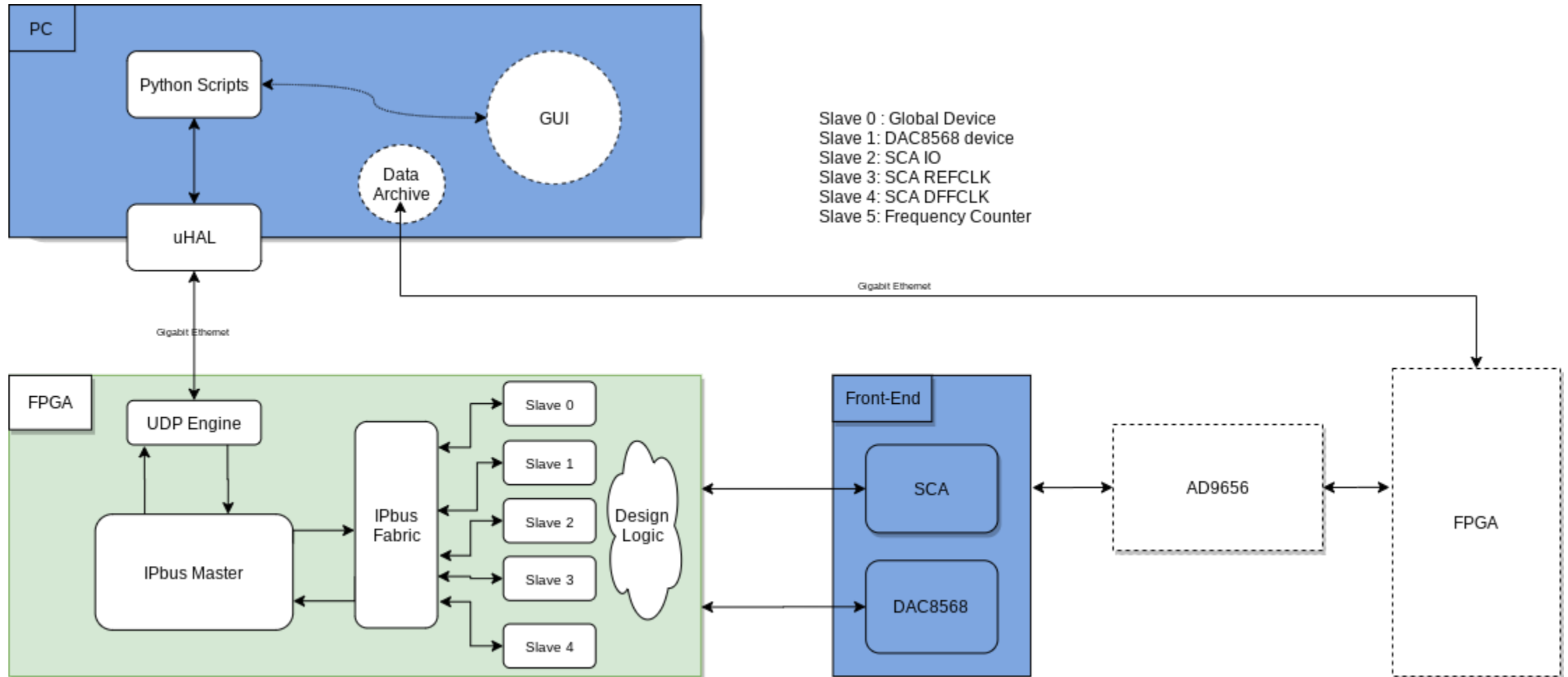
- Ubuntu 20.04 LTS
- Xilinx Vivado 2019.2
- Python 3.8.2
- IPbus Software: master branch
- IPbus Firmware: master branch

For the PC connected to FPGA:

- Most Linux distributions (Centos/Debian/Ubuntu..) are supported.
- **Vivado Lab Edition(2019.2)** should be installed to program FPGA.
- IPbus Software build with Python3

Data Readout

Let's use AD9656



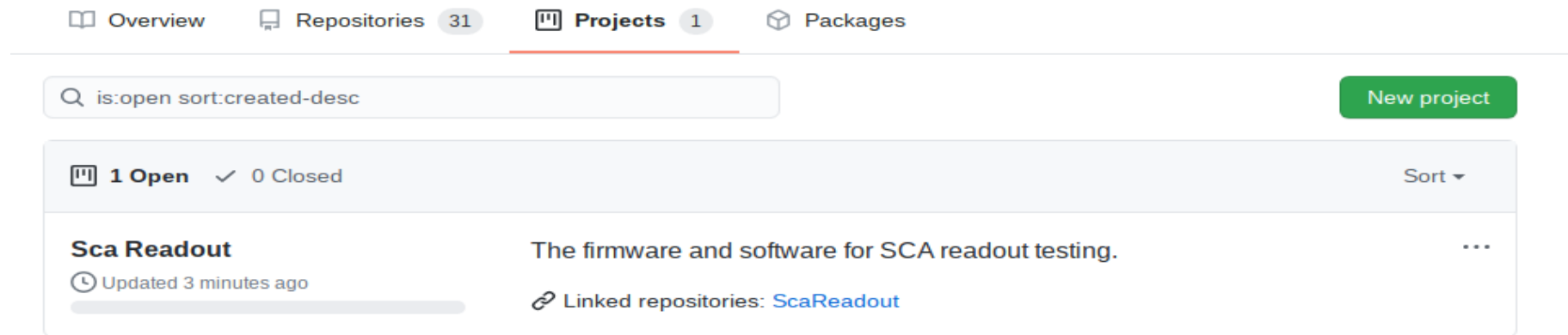
Repositories

Github Repo:

<https://github.com/habrade/ScaReadout>

Github Project:

Do we need this, or we just use wiki?



The screenshot shows the GitHub Projects interface. At the top, there are navigation tabs: Overview, Repositories (31), Projects (1), and Packages. The Projects tab is selected and underlined. Below the tabs is a search bar containing the text "is:open sort:created-desc" and a green "New project" button. The main content area shows a summary of projects: "1 Open" and "0 Closed". A single project is listed: "Sca Readout" with a description "The firmware and software for SCA readout testing." and a status "Updated 3 minutes ago". Below the description, it says "Linked repositories: ScaReadout".